16.看门狗

1．实验目的

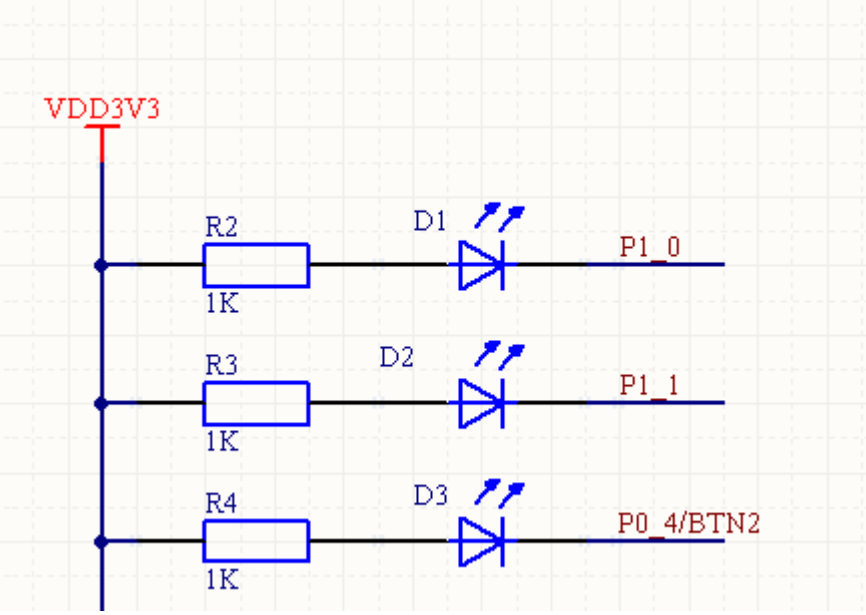
看门狗是在软件跑飞的情况下 CPU 自恢复的一个方式，当软件在选定的时间间隔内不能置位看门狗定时器(WDT)，WDT 就复位系统。看门狗可用于电噪声，电源故障或静电放电等恶劣工作环境或高可靠性要求的环境。如果系统不需要应用到看门狗，则 WDT 可配置成间隔定时器，在选定时间间隔内产生中断。WDT 的特性如下：4 个可选择的时间间隔看门狗定时器模式下产生中断请求时钟独立于系统时钟，WDT 包括一个 15 位定时/计数器，它的频率由 32.768KHz 的晶振决定。用户不能查看计数器的值工作于各个电源模式。让用户了解几种看门狗定时器的使用。

2．实验设备

硬件：PC 机一台 ZB 网关（底板、核心板、仿真器、USB 线）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

3．实验相关电路图



4．实验分析及相关寄存器

相关寄存器 WDCTL 如下表所示：(详细参考 CC2530 中文数据手册完全版.pdf)

按照表格寄存器内容，我们对 WDCTL 具体配置可如下：

寄存器	位	描述
WDCTL	Bit[7:4]	清除定时器。当 0xA 跟随 0x5 写到这些位，定时器被清除（即加载 0）。注意定时器仅写入 0xA 后，在 1 个看门狗时钟周期内写入 0x5 时被清除。当看门狗定时器是 IDLE 为时写这些位没有影响。当运行在定时器模式，定时器可以通过写 1 到 CLR[0]（不管其他 3 位）被清除为 0x0000（但是不停止）。
	Bit[3:2]	模式选择。该位用于启动 WDT 处于看门狗模式还是定时器模式。当处于定时器模式，设置这些位为 IDLE 将停止定时器。注意：当运行在定时器模式时要转换到看门狗模式，首先停止 WDT，然后启动 WDT 处于看门狗 模式。当运行在看门狗模式，写这些位没有影响。 00：IDLE 01：IDLE（未使用，等于 00 设置） 10：看门狗模式 11：定时器模式
	Bit[1:0]	定时器间隔选择。这些位选择定时器间隔定义为 32 kHz 振荡器周期的规定数。注意间隔只能在 WDT 处于 IDLE 时改变，这样间隔必须在定时器启动的同时设置。 00：定时周期×32,768（~1 s）当运行在 32 kHz XOSC 01：定时周期×8192（~0.25 s） 10：定时周期×512（~15.625 ms） 11：定时周期×64（~1.9 ms）

WDCTL = 0x00; //打开 IDLE 才能设置看门狗
WDCTL |= 0x08; //定时器间隔选择,间隔一秒

停止喂狗：
WDCTL = 0xa0; //清除定时器。当 0xA 跟随 0x5 写到这些位，定时器被清除
WDCTL = 0x50;

5．源码分析

```
# include <ioCC2530.h>
#define uint unsigned int
#define LED1 P1_0
#define LED2 P1_1
void InitLEDIO(void)
{
P1DIR |= 0x03; //P10、P11 定义为输出
LED1 = 1;
LED2 = 1;
//LED 灯初始化为关
}
void Init_Watchdog(void)
{
WDCTL = 0x00;
//时间间隔一秒，看门狗模式
WDCTL |= 0x08;
//启动看门狗
}
void SET_MAIN_CLOCK(source)
{
if(source) {
CLKCONCMD |= 0x40; /*RC*/
while(!(CLKCONSTA &0X40)); /*待稳*/
}
else {
CLKCONCMD &= ~0x47; /*晶振*/
while((CLKCONSTA &0X40)); /*待稳*/
}
}
void FeetDog(void)
```

```
{
WDCTL = 0xa0;
WDCTL = 0x50;
}
void Delay(uint n)
{
uint i;
for(i=0;i<n;i++);
for(i=0;i<n;i++);
for(i=0;i<n;i++);
for(i=0;i<n;i++);
for(i=0;i<n;i++);
}
void main(void)
{
SET_MAIN_CLOCK(0) ;
InitLEDIO();
Init_Watchdog();
Delay(10000);
LED1=0;
LED2=0;
while(1)
{
//FeetDog();
} //喂狗指令（加入后系统不复位，LED1 和 LED2 不再闪烁）
}
```

实验效果

1.

```
LED1=0;
LED2=0;
while(1)
{
        //FeetDog();
}       //喂狗指令（加入后系统不复位，LED1和LED2不再闪烁）
```

代码如上配置，LED1 LED2 不断闪烁，系统不断复位。

2.

```
LED1=0;
LED2=0;
while(1)
{
        FeetDog();
}       //喂狗指令（加入后系统不复位，LED1和LED2不再闪烁）
```

代码如上配置，LED1 LED2 不闪烁 ，系统不复位了。

技术工程师: QQ 2357481431
公司网站：http://www.csgsm.com 店铺：http://520ic.taobao.com/